I chose to look at the city of Regina in Saskatchewan, Canada because Saskatoon is my hometown but that city is too small.

Contents

Problems Encountered in the Map

Investigating element tags

Running the mapparser.py file shows that there are the following tags:

```
{'bounds': 1,
'member': 4816,
'nd': 302102,
'node': 299241,
'osm': 1,
'relation': 78,
'tag': 187219,
'way': 36611}
```

I know that the one `bound` tag indicates the bounding limits of the city data and the `osm` tag is present at the beginning of all OSM files. The three OSM data primitives (nodes, ways, and relations) are present as `node`, `way`, and `relation`. There are some super relations, such as the Trans-Canada Highway, which is a major highway that runs east to west through Canada and passes through Regina. The `nd` tags are children of `way` elements and the `member` tags are children of `relation` elements. The `tag` tag can be found under any of the three data primitives. There are no unexpected tags and the relative quantities of all the tags seems reasonable (e.g. less `tags` than `nodes` and `ways`).

This file does not have the attribute `visible` for any elements and so I have omitted the `visible` key:value pair in node dict when parsing the file.

Investigating tag element ‘k’ values

From tag.py the results are:

- 64063 tag `'k'` values with only lower case letters and underscores
- 122354 tag `'k'` values that are only lower case letters and underscores with a colon in the middle
- 802 other tag `'k'` values
- 0 problem character tag `'k'` values

I was curious what the specific tag `'k'` values were so I modified the mapparser.py file to create a new file (tagparser.py) to count the different tag `'k'` values in the three different categories and discovered a few things:

Other tag ‘k’ values

All of the other tag `'k'` values are `NHS`, `UNIT`, `VACANT`, or `subdi1name`; none of these are tags I feel should ultimately be included in the data so when parsing the OSM file to JSON I will skip any tag with these values.

Colon tag ‘k’ values

Within the colon tag `'k'` values there are 8 address values with the following counts:

```
{'addr:city': 232,
 'addr:country': 4,
 'addr:housename': 12,
 'addr:housenumber': 60905,
 'addr:postcode': 111,
 'addr:province': 10,
 'addr:street': 60988,
 'addr:unit': 1…
```

- From the numbers we can see that city, province, and country are not included in most addresses and so for the purposes of standardizing the data I will omit these values when parsing the file. A `housename` value is only given for 12 addresses so I will not include these either.
- The values of `housenumber` and `street` are not the same; therefore, there must be 83 addresses that only have a `street` value and not a `housenumber`.
- Not all addresses have a postal code, but I will leave these in because postal codes are the most accurate identifiers of a building’s location.

Lower case tag ‘k’ values

Contained in lowerlog.txt. All of the lower case appear to be miscellaneous information that occurs relatively infrequently does not appear to be problematic (although it may not be particularly useful). I will include all of these as a regular key:value pairs.

Investigating ‘addr:street’ values

From audit.py the results showed several things:

- There are several common valid street types that were not in the expected list such as: Crescent and Way. There are also many valid street types which appeared less commonly.
    - These valid street types were added to the expected list
    - After this there were a few unexpected street types such as all caps names or abbreviated names (e.g. BOULEVARD). These unexpected street types were added to the mapping dict. It is unlikely that only the street type is all caps so I have added the `.title()` method when correcting street names to fix the rest of the name as well
- A direction is often placed after the street name; this either signifies the quadrant of the city the street is located in or the direction of traffic flow. Because these direction are typically displayed as a single letter (e.g. N for North) in actual road signs I will stick with the single letter convention rather than spelling out the full word.

Investigating ‘addr:postcode’ values

All postal codes in Regina should begin with the S4. The third letter can be: L, M, N, P, R, S, T, V, W, X, Y, or Z. I used address.py to check this and all postal codes present are valid.

Investigating ‘addr:housenumber’ values

I used housenumber.py to check if all `housenumber` values were numbers and got the following results:

```
{'2A': 2,
'2B': 1,
'3A': 1,
'3B': 1,
'440A': 1,
'440B': 1,
'49-2223': 1,
'663A': 1,
'663B': 1,
'7A': 2,
'7B': 2,
'8A': 1,
'9A': 1,
'9B': 1,
'Northgate Mall Food Court': 1,
'Southland Mall Food Court': 1,
'Victoria Square Mall': 1}
```

The last three values are not valid house numbers and should be omitted. The other values that are alphanumeric are most likely unit numbers for apartments rather than actual house numbers. The one hyphenated value is probably a combination of unit number and building number.

Overview of Data

I used `mongoimport` to upload my new file regina_canada.osm.json into a collection called `osm` in the `examples` database.

Size of file

```
regina_canada.osm = 65 MB
regina_canada.osm.json = 74 MB
```

Unique users

```
>> db.osm.aggregate([
     {"$group": {"_id": "$created.user", "count":{"$sum":1}}},
     {"$sort": {"count":-1}}])

"_id" : "reginaab", "count": 190209
"_id" : "gecho111", "count": 108298
"_id" : "Geospizinae", "count": 10485…
```

The top contributor is `reginaab`, who has contributed 57% of documents.

Number of nodes

```
>> db.osm.find({"type":"node"}).count()
299241
```

Number of ways

```
>> db.osm.find({"type":"node"}).count()
36601
```

Religion

```
>> db.osm.aggregate([
     {"$group": {"_id": "$religion", "count":{"$sum":1}}},
     {"$sort": {"count":-1}}])

"_id" : null, "count": 335801
"_id" : "christian", "count": 49
"_id" : "hindu", "count": 1
"_id" : "jewish", "count": 1
```

It is not surprising that Christianity seems to be prevalent religion given that the 2011 Canadian Census showed that top three responses for religion in Regina were Protestant (41.5%), Roman Catholic (32.3%), and no religion (19.0%).

Buildings

The `building` tag is the most common tag (24650 instances from tagparser.py). Investigating the values for this tag further shows:

```
>> db.osm.aggregate([
     {"$group": {"_id": "$building", "count":{"$sum":1}}},
     {"$sort":{"count":-1}}])

"_id" : "null", "count": 318912
"_id" : "yes", "count": 15701
"_id" : "commercial", "count": 313
"_id" : "apartments", "count": 567
"_id" : "house", "count": 109
"_id" : "industrial", "count": 71
"_id" : "school", "count": 62
"_id" : "retail", "count": 39
"_id" : "storage_tank", "count": 29
"_id" : "university", "count": 18
"_id" : "residential", "count": 12
"_id" : "public", "count": 6
"_id" : "om", "count": 4
"_id" : "office", "count": 3
"_id" : "church", "count": 2
"_id" : "garage", "count": 2
"_id" : "roof", "count": 1
"_id" : "hotel", "count": 1
```

Most `building` tags simply have the value `"yes"`, which is not very descriptive. Some values do not make sense i.e. `"om"`, `"roof"`, and possibly `"storage_tank"`. The distinctions between what would qualify as a `"residential"` building versus `"apartments"` and `"house"` are unclear; similarly, the distinctions between `"commercial"`, `"office"`, and `"industrial"` are also unclear.

Other Ideas

89% of the documents submitted were submitted by the top two contributors: `reginaab` and `gecho111`. It's clear that much of the information is missing altogether (as in the case of postal codes) or lacking in specificity (tag values for nodes). The main industries in Saskatchewan are natural resources (potash, oil and gas, agriculture, etc.). OpenStreetMaps is not likely to be on the radar for most residents and as such the pool of people who are making contributions is very small (shown by the fact that almost 90% of data is contributed by only two users). It's very difficult for only two users to contribute specific information about what is specifically located at each node. Major facilities such as some university buildings and the recycling depot have been included; but, the majority of businesses/amenities are predictably left out.

Encouraging business owners to add their own data would improve the usefulness of the Regina OSM greatly. Business owners may not gain that much by adding their business information to OSM if the pool of people who use the site is small. However, if someone were to use it and see that there is only one medical clinic (i.e. Quance East Medical Clinic) that clinic would essentially have a monopoly on the – albeit likely small – group of OSM users. I think the best way to add this data would be to add more complete postal code data to the map (postal code to location data from Canada Post) and then mapping local business directory data to map using postal codes, rather than addresses.

However, I do not believe that there is a dataset coordinating postal codes with GPS coordinates readily available for download by Canada Post. One has to query their website with an address and receive a postal code in turn. This process could likely be automated but obtaining a postal code requires knowledge of the address in the first place. In addition, there is no check for the correct GPS location. If the address is located at the wrong coordinates, a postal code could still be obtained but now the address, along with postal code, would be simply be located at the wrong place. This does not necessarily improve the map as a whole.

There is one way in which I think OSM could be more useful in more rural areas in Canada by allowing individuals to add points in remote locations (farms, industrial plant sites, etc.) that do not have traditional street addresses. These locations are typically referred to by a group of numbers known as a LSD and LSDs cannot be searched directly on any of the popular maps. LSD apps that convert LSDs to GPS coordinates do exist but in my experience they are typically not very precise as they can only narrow down the point to one section (which is still a relatively large area).